Lecture Notes in Artificial Intelligence 12833

Subseries of Lecture Notes in Computer Science

More information about this subseries at http://www.springer.com/series/1244

Fairouz Kamareddine · Claudio Sacerdoti Coen (Eds.)

Intelligent Computer Mathematics

14th International Conference, CICM 2021
Timisoara, Romania, July 26–31, 2021
Proceedings

Springer

Editors
Fairouz Kamareddine
Heriot-Watt University
Edinburgh, UK

Claudio Sacerdoti Coen
University of Bologna
Bologna, Italy

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-030-81096-2 ISBN 978-3-030-81097-9 (eBook)
https://doi.org/10.1007/978-3-030-81097-9

LNCS Sublibrary: SL7 – Artificial Intelligence

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

With the continuing, rapid progress of digital methods in communications, knowledge representation, processing, and discovery, the unique character and needs of mathematical information require unique approaches. Its specialized representations and capacity for creation and proof, both automatically and formally as well as manually, set mathematical knowledge apart.

The Conference on Intelligent Computer Mathematics (CICM) was initially formed in 2008 as a joint meeting of communities involved in computer algebra systems, automated theorem provers, and mathematical knowledge management, as well as those involved in a variety of aspects of scientific document archives. It has offered a venue for discussing, developing, and integrating the diverse, sometimes eclectic, approaches and research. Since 2008, CICM has been held annually: Birmingham (UK, 2008), Grand Bend (Canada, 2009), Paris (France, 2010), Bertinoro (Italy, 2011), Bremen (Germany, 2012), Bath (UK, 2013), Coimbra (Portugal, 2014), Washington D. C. (USA, 2015), Bialystok (Poland, 202016), Edinburgh (UK, 2017), Linz (Austria, 2018), Prague (Czech Republic, 2019) and Bertinoro (Italy, 2020). This latter edition, which was originally scheduled to be held in Bertinoro, Italy, was hosted online due to the COVID-19 pandemic. This year's meeting was supposed to be held in Timisoara, Romania, but again due to the pandemic, it was held online (July 26–31, 2021).

This year's meeting exposed advances in formalizations, automatic theorem proving, applications of machine learning to mathematical documents and proof search, search and classifications of mathematical documents, teaching and geometric reasoning, and logic and systems, among other topics. This volume contains the contributions to this conference. From 38 formal submissions, the Program Committee (PC) accepted 20 papers including 12 full research papers, 7 shorter papers describing software systems or datasets and 1 paper highlighting development of systems and tools in the last year. All papers were reviewed by at least three PC members or external reviewers. The reviews were single-blind and included a response period in which the authors could respond and clarify points raised by the reviewers. In addition to the main sessions, the conference included a doctoral program, chaired by Yasmine Sharoda, which provided a forum for PhD students to present their research and get advice from senior members of the community. Additionally, the following workshops were scheduled:

- The 31st OpenMath Workshop, organized by James Davenport and Michael Kohlhase.
- The 2nd Workshop on Natural Formal Mathematics (NatFoM 2021), organized by Peter Koepke and Dennis Müller.
- The 5th Workshop on Formal Mathematics for Mathematicians (FMM 2021), organized by Jasmine Blanchette and Adam Naumowicz.

- The 2nd Workshop on Formal Verification of Physical Systems (FVPS 2021), organized by Sofiene Tahar, Osman Hasan and Adnan Rashid.
- The 13th Workshop on Mathematical User Interaction (MathUI 2021), organized by Andrea Kohlhase.

Finally, the conference included four invited talks:

- Alessandro Cimatti (Fondazione Bruno Kessler, Italy): “Logic at work, and some research challenges for computer mathematics”.
- Michael Kohlhase (FAU Erlangen-Nürnberg, Germany): “Referential Semantics – a Concept for Bridging between Representations of mathematical/technical Documents and Knowledge”.
- Laura Kovacs (TU Vienna, Austria): “Induction in Saturation-Based Reasoning”.
- Angus McIntyre (Emeritus Professor, Queen Mary University of London, UK): “Doing classical number theory in weak axiomatic systems”.

A successful conference is due to the efforts of many people. We thank Madalina Erascu and her colleagues at the West University of Timisoara, Romania, for the difficult task of organizing a conference with the expectation of it being held face to face but with the dynamics of COVID-19 making it difficult to accommodate in person meetings. We are grateful to Serge Autexier for his publicity work. We also thank the authors of submitted papers, the PC for their reviews, and the organizers of the workshops, as well as the invited speakers and the participants of the conference.

June 2021

F. Kamareddine
C. Sacerdoti Coen

Organization

Program Committee Chairs

Fairouz Kamareddine	Heriot-Watt University, UK
Claudio Sacerdoti Coen	University of Bologna, Italy

Program Committee

Akiko Aizawa	National Institute of Informatics, Japan
Mauricio Ayala-Rincón	Universidade de Brasilía, Brasil
Frédéric Blanqui	Inria, France
Jacques Carette	McMaster University, Canada
Howard Cohl	NIST, USA
James H. Davenport	University of Bath, UK
Catherine Dubois	ENSIIE-Samovar, France
Jacques Fleuriot	The University of Edinburgh, UK
Osman Hasan	National University of Sciences and Technology, Pakistan
Jan Jakubuv	Czech Technical University, Czech Republic
Mateja Jamnik	University of Cambridge, UK
Moa Johansson	Chalmers University of Technology, Sweden
Cezary Kaliszyk	University of Innsbruck, Austria
Manfred Kerber	University of Birmingham, UK
Andrea Kohlhase	University of Applied Sciences Neu-Ulm, Germany
Adam Naumowicz	Institute of Informatics, University of Bialystok, Poland
Olga Nevzorova	Kazan Federal University, Russia
Markus N. Rabe	Google, USA
Florian Rabe	FAU Erlangen-Nürnberg, Germany
Moritz Schubotz	Universität Konstanz, Germany
Stephan Schulz	DHBW Stuttgart, Germany
Volker Sorge	University of Birmingham, UK
Olaf Teschke	FIZ Karlsruhe, Germany
Joe Wells	Heriot-Watt University, UK
Makarius Wenzel	sketis.net, Germany
Wolfgang Windsteiger	RISC Institute, Austria
Richard Zanibbi	Rochester Institute of Technology, USA

Additional Reviewers

Thaynara Arielly de Lima
Ciaran Dunne
André Greiner-Petter
Alexander Kirillovich
Artur Korniłowicz
Dennis Müller
Miroslav Olšák
Stanisław Purgał
Adnan Rashid
Max Rapp
Colin Rothgang
Philipp Scharpf
Jonas Schopf
Philipp Smola
René Thiemann
Josef Urban

Invited Talks

Logics at Work, and Some Challenges for Computer Mathematics

Alessandro Cimatti

Fondazione Bruno Kessler
cimatti@fbk.eu

Formal verification aims at the exhaustive analysis of the behaviours of a system, to ensure that the expected properties are universally met. Formal verification has been applied in many sectors including control software, relay interlocking, space, avionics, hardware circuits, and production plants. We informally distinguish systems in discrete systems and hybrid systems.

Symbolic Verification of Transition Systems. In case of discrete systems, a behaviour can be seen as a sequence of valuations to a set of state variables. We focus on symbolic verification, where logical methods are used to represent and explore the system model. In the case of transition systems, a state is represented as an assignment to a set of logical variables V. Logical formulae are used to represent sets of states, so that $I(V)$ represents the (initial) states satisfying I, and $T(V, V')$ represents sets of transitions, with V' being the next state variables. In the finite-state case, V is a vector of Boolean variables. Symbolic algorithms for automated verification [8], originally based on Binary Decision Diagrams [7], have progressively been replaced by verification based on satisfiability checking (SAT) [5]. SAT-based model checking techniques include Bounded Model Checking [4], induction [19], interpolation [17] and IC3 [6].

In case of infinite-state transition systems, the state variables may have infinite range, and I and T are generally expressed as formulae in first-order logic, in the framework of Satisfiability Modulo Theories (SMT) [3]. SMT extends the propositional case by allowing for functions and relations between individual variables, with interpretations over relevant theories. These include linear and non-linear real and integer arithmetic (LRA, NRA, LIA, NIA). The algorithms for the analysis of infinite-state transition systems, also referred to as Verification Modulo Theories [11], are not only obtained by replacing SAT solvers with SMT solvers in SAT-based verification approaches [2, 14]. A fundamental role is played by abstractions, most notably predicate abstraction [16]. Abstractions are dynamically refined based on the analysis of abstract counterexamples [13], and can be either computed explicitly, or implicitly [20], in tight integration with verification algorithms such as IC3 [10].Particularly relevant for non-linear theories is the case of incremental linearization [9], where the abstract space is built by treating non-linearities as uninterpreted functions with piecewise-linear bounds.

Verification of Hybrid Systems. In the case of continuous time, the situation is significantly more complex. In fact, hybrid systems are composed of interacting discrete and continuous subsystems. Within the reference modeling framework of Hybrid Automata [1], two kinds of transitions exist: discrete transitions, where the system instantaneously switches from a discrete mode to the next, and continuous transitions, where time elapses while in one mode, with continuous variables evolve according to the specified laws. With respect to the case of transition systems, the semantics of hybrid automata comes with an implicit elapse of time, during which continuous variables evolve according to specific laws defined by differential equations, subject to invariants that must hold throughout the continuous transitions. The traditional approaches are based on an explicit enumeration of the modes and the analysis of the differential equations in the various modes. We focus on symbolic, logic-based approaches [12, 15, 18], where deductive methods are used to analyze the continuous dynamics.

In this setting, we can identify several interesting challenges.

Satisfiability Modulo Theories. At the level of SMT engines, a key problem is to provide efficient and effective theory solvers for non-linear theories, to be integrated within the standard online SMT search schema [3]. In addition to incrementality and the ability to construct theory lemmas, a fundamental requirement could be referred to as “non-constructive satisfiability”, i.e. the ability to prove the satisfiability of a set of constraints without actually having to produce a model. Algorithms for incomplete theory reasoning, providing efficiently sufficient conditions for satisfiability or for unsatisfiability, would also be very useful.

Verification Modulo Theories. At the level of verification of transition systems over non-linear and transcendental theories, most techniques are oriented to prove universal properties, whereas existential properties have been devoted less attention. Non-constructive satisfiability would be an important tool in abstraction refinement, to prove the existence of infinite behaviours. In fact, traces can not be finitely presented in lasso-shape form as for the finite-state case.

Hybrid Automata. A key challenge is to integrate within the existing symbolic algorithms the large body of work on characterizing, checking, and finding differential invariants that has been developed in the setting of dynamical systems. Depending on the nature of the system (e.g. linear, non-linear polynomial, or featuring transcendental functions), different invariants could be found (e.g. polynomial equalities and inequalities).

In some cases, hybrid automata can be reduced to the analysis of infinite-state transition systems, so that the SMT-based approaches can be leveraged. Such precise encodings rely on the existence of a closed-form exact solution. Even in such subcases, an important challenge is to improve the quantifier-free encoding of invariant conditions [12].

Finally, it would be interesting to support the direct reasoning at the level of differential equations in the verification algorithm, in the style of [15], but to more advanced algorithms such as IC3. The requirement is to identify procedures for the checks of induction (and relative induction) under the differential equations.

References

1. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.H.: Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (eds.) HS 1992, HS 1991. LNCS, vol. 736, pp. 209–229. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57318-6_30
2. Audemard, G., Bozzano, M., Cimatti, A., Sebastiani, R.E.: Verifying industrial hybrid systems with MathSAT. Electron. Notes Theor. Comput. Sci. **119**(2), 17–32 (2005)
3. Barrett, C.W., Sebastiani, R., Seshia, S.A., Tinelli, C.: Satisfiability modulo theories. In: Handbook of Satisfiability, vol. 185. Frontiers in AI and Applications, pp. 825–885. IOS Press (2009)
4. Biere, A., Cimatti, A., Clarke, E., Zhu, Y.: Symbolic model checking without BDDs. In: Cleaveland, W.R. (eds.) TACAS 1999. LNCS. Vol. 1579, pp. 193–207. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49059-0_14
5. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability, vol. 185. Frontiers in AI and Applications. IOS Press (2009)
6. Bradley, A.R.: SAT-Based model checking without unrolling. In: Jhala, R., Schmidt, D. (eds.) VMCAI 2011. LNCS, vol. 6538, pp. 70–87. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18275-4_7
7. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Trans. Comput. 35(8), 677–691 (1986)
8. Burch, J., Clarke, E.M., McMillan, K.L., Dill, D.L., Hwang, L.J.: Symbolic model checking: 10ˆ20 states and beyond. Inf. Comput. **98**(2), 142–170 (1992)
9. Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Incremental linearization for satisfiability and verification modulo nonlinear arithmetic and transcendental functions. ACM Trans. Comput. Log. **19**(3), 19:1–19:52 (2018)
10. Cimatti, A., Griggio, A., Mover, S., Tonetta, S.: Infinite-state invariant checking with IC3 and predicate abstraction. Formal Methods Syst. Des. **49**(3), 190–218 (2016). https://doi.org/10.1007/s10703-016-0257-4
11. Cimatti, A., Griggio, A., Tonetta, S. Verification modulo theories: language, benchmarks and tools (2011). http://vmt-lib.fbk.eu/
12. Alessandro Cimatti, Sergio Mover, and Stefano Tonetta. Quantifier-free encoding of invariants for hybrid systems. Formal Methods Syst. Des. **45**(2), 165–188, 2014.
13. Clarke, E.M., Grumberg, O., Jha, S., Lu, Y., Veith, H.. Counterexample-guided abstraction refinement for symbolic model checking. J. ACM, **50**(5), 752–794 (2003)
14. de Moura, L., Rueß, H., Sorea, M.: Lazy theorem proving for bounded model checking over infinite domains. In: Voronkov, A. (eds.) CADE 2002. LNCS, vol. 2392, pp. 438–455. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45620-1_35
15. Eggers, A., Fränzle, M., Herde, C.: SAT modulo ODE: a direct SAT approach to hybrid systems. In: Cha, S.., Choi, J.Y., Kim, M., Lee, I., Viswanathan, M. (eds.) ATVA 2008. LNCS, vol. 5311, pp. 171–185. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88387-6_14
16. Graf, S., Saidi, H.: Construction of abstract state graphs with PVS. In: Grumberg, O. (eds.) CAV 1997. LNCS, vol. 1254, pp. 72–83. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63166-6_10
17. McMillan, K.L.: Interpolation and model checking. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (eds.) Handbook of Model Checking, pp. 421–446. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-10575-8_14

18. Platzer, A.: Logical Foundations of Cyber-Physical Systems. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-63588-0
19. Sheeran, M., Singh, S., Stålmarck, G.: Checking safety properties using induction and a SAT-solver. In: Hunt, W.A., Johnson, S.D. (eds.) FMCAD 2000. LNCS, vol. 1954, pp. 108–125. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-40922-X_8
20. Tonetta, S.: Abstract Model checking without computing the abstraction. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 89–105. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05089-3_7

Induction in Saturation-Based Reasoning

Laura Kovács

TU Wien, Austria
laura.kovacs@tuwien.ac.at

Keywords: Automated reasoning · Theorem proving · Induction

Extended Abstract

Seminal works on automating induction mainly focus on inductive theorem proving [1, 2]: deciding when induction should be applied and what induction axiom should be used. Further restrictions are made on the logical expressiveness, for example induction over only universal properties [1, 13] and without uninterpreted symbols [10], or only over term algebras [5, 8]. Inductive proofs usually rely on auxiliary lemmas to help proving an inductive property. In [3] heuristics for finding such lemmas are introduced, for example by randomly generating equational formulas over random inputs and using these formulas if they hold reasonably often. Recent advances related to automating inductive reasoning, such as first-order reasoning with inductively defined data types [9], inductive strengthening of SMT properties [12], structural induction in superposition [4, 5, 6, 8, 11], open up new possibilities for automating induction. In this talk, we describe our extensions to first-order theorem proving in support of automating inductive reasoning.

It is common in inductive theorem proving, that given a formula/goal F, try to prove a more general goal instead [1, 2]. Such an approach however does not apply in the context of saturation-based first-order theorem proving, which is not based on a goal-subgoal architecture. In our work we therefore integrate induction directly into saturation-based proof search. We do so by turning applications of induction into inference rules of the saturation process and adding instances of appropriate induction schemata. To this extent, we pick up a formula F in the search space and add to the search space new induction axioms, that is instances of induction schemata, aiming at proving $\neg F$, or sometimes even a more general formula than $\neg F$. Our recent works [6, 7] investigated such an approach, introducing new inference rules for induction in saturation-based first-order theorem proving.

Our inference rules for induction in saturation capture the application of induction to inductive formulas to be proved. However, this is insufficient for efficient theorem proving. Modern saturation-based theorem provers are very powerful not just because of the logical calculi they are based on, such as superposition. What makes them powerful and efficient are (i) redundancy criteria and pruning search space, (ii) strategies for directing proof search, mainly by clause and inference selection, and recent results on (iii) theory-specific reasoning, for example with inductive data types. We overview our results in mechanizing mathematical induction in saturation-based first-order theorem proving in an efficient way. In particular we describe induction in saturation by generalizing inductive formulas [6] with/without recursive functions and integers [7].

Acknowledgements. The results described in this talk are based on joint works with Márton Hajdú, Petra Hozzvá, Johannes Schoisswohl and Andrei Voronkov. We acknowledge funding from the ERC CoG ARTIST 101002685, the ERC StG 2014 SYMCAR 639270, the EPSRC grant EP/P03408X/1 and the Austrian FWF research project LogiCS W1255-N23.

References

1. Boyer, R.S., Moore, J.S.: A Computational Logic Handbook, Perspectives in Computing, vol. 23. Academic Press (1979)
2. Bundy, A., Stevens, A., van Harmelen, F., Ireland, A., Smaill, A.: Rippling: a heuristic for guiding inductive proofs. Artif. Intell. **62**(2), 185–253 (1993)
3. Claessen, K., Johansson, M., Rosén, D., Smallbone, N.: HipSpec: automating inductive proofs of program properties. In: Proceedings of the ATx/WinG, pp. 16–25 (2012)
4. Cruanes, S.: Superposition with structural induction. In: Dixon, C., Finger, M., (eds.) FroCoS 2017. LNCS, vol. 10483, pp. 172-188. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66167-4_10
5. Echenheim, M., Peltier, N.: Combining induction and saturation-based theorem proving. J. Autom. Reason. 64, 253–294 (2020). https://doi.org/10.1007/s10817-019-09519-x
6. Hajdú M., Hozzová, P., Kovács, L., Schoisswohl, J., Voronkov, A.: Induction with generalization in superposition reasoning. In: Benzmüller, C., Miller, B. (eds.) CICM 2020. LNCS, vol 12236, pp. 123–137. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53518-6_8
7. Hozzová, P., Kovács, L., Voronkov, A.: Integer induction in saturation. In: Proceedings of the CADE (2021, to appear)
8. Kersani, A., Peltier, N.: Combining superposition and induction: a practical realization. In: Fontaine, P., Ringeissen, C., Schmidt, R.A. (eds.) FroCoS 2013. LNCS, vol. 8152, pp. 7–22. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40885-4_2
9. Kovács, L., Robillard, S., Voronkov, A.: Coming to terms with quantified reasoning. In: Proceedings of the POPL. pp. 260–270 (2017)
10. Passmore, G., et al.: The Imandra automated reasoning system (system description). In: Peltier, N., Sofronie-Stokkermans, V. (eds.) IJCAR 2020. LNCS, vol. 12167, pp. 464–471. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51054-1_30
11. Reger, G., Voronkov, A.: Induction in saturation-based proof search. In: Fontaine, P., (eds.) CADE 2019. LNCS, vol. 11716, pp. 477–494. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29436-6_28
12. Reynolds, A., Kuncak, V.: Induction for SMT solvers. In: D'Souza, D., Lal, A., Larsen, K.G. (eds.) VMCAI 2015. LNCS, vol. 8931, pp. 80–98. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46081-8_5
13. Sonnex, W., Drossopoulou, S., Eisenbach, S.: Zeno: an automated prover for properties of recursive data structures. In: Flanagan, C., König, B. (eds.) TACAS 2012. LNCS, vol. 7214, pp. 407–421. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28756-5_28

Doing Number Theory in Weak Systems of Arithmetic

Angus Macintyre

University of Edinburgh
a.macintyre@qmul.ac.uk

Abstract. Although Godel's Theorem shows that even ZFC is incomplete for unsolvability of diophantine equations, nothing explicit of any real interest to number theorists has ever been shown to be unprovable. I will consider various important statements about solvability modulo all prime powers, and exhibit a wide class which get decided by PA (first order Peano Arithmetic) using serious algebraic geometry inside nonstandard models of PA. So although PA is often misrepresented as very weak, it is rather strong for basic results of 20th century number theory.

Keywords: Number theory · Weak arithmetic

Contents

Search and Classification

Teaching and Geometric Reasoning

Logic and Systems